



## 3. Aufgabenblatt

31.10.2013

Die Aufgaben der Präsenzübung sollen in den Übungsgruppen bearbeitet werden. Die Hausaufgaben sind **vom 07.11. bis 13.11.** bei Ihrem jeweiligen Tutor zu Ihrer Präsenzübungszeit in handschriftlicher Form abzugeben. Die Abgaben sollen alle zum Nachvollziehen der Lösungen notwendigen Zwischenschritte, sowie den Namen des Tutors und die Übungsgruppe enthalten. Die Abgaben erfolgen in Teams mit drei Personen, wobei alle Mitglieder des Teams derselben Übungsgruppe angehören müssen. Jedes der Mitglieder muss eine eigene, handschriftliche Lösung abgeben und die anderen Mitglieder sind auf den Lösungen deutlich anzugeben.

### Aufgabe 1: Schnelltest

1. Welche der folgenden Aussagen zur FPU sind korrekt?
  - ☐ Die FPU-Register sind als Stack angeordnet.
  - ☐ Die FPU-Register sind breiter als die regulären Register.
  - ☐ Float Werte können nicht als globale Variablen im `.data` Teil abgelegt werden.
  - ☐ Über `%st(1)` greift man immer auf das selbe FPU-Register zu.
2. Wo wird bei einem Funktionsaufruf durch den Befehl `call` die Rücksprungadresse gesichert?
  - ☐ Stack
  - ☐ Im `(.data)` Teil
  - ☐ Register
  - ☐ Festplatte / Massenspeicher
3. Wo werden globale Variablen die im `.data` Teil deklariert werden abgelegt?
  - ☐ Stack
  - ☐ Arbeitsspeicher
  - ☐ Register
  - ☐ Festplatte / Massenspeicher
4. Wo können Sie Übergabeparameter an die aufgerufene Funktion übergeben werden sollen ablegen?
  - ☐ Stack
  - ☐ Im `(.data)` Teil
  - ☐ Register
  - ☐ Arbeitsspeicher
5. Was macht der Befehl `leal a, %esi`?
  - ☐ Die Adresse von `%esi` wird in `a` gespeichert.
  - ☐ Der Wert, der in `a` gespeichert ist, wird als Adresse interpretiert und der Wert an dieser Adresse wird in `%esi` gespeichert.
  - ☐ Die Adresse von `a` wird in `%esi` gespeichert.
  - ☐ Der Wert, der in `%esi` gespeichert ist, wird als Adresse interpretiert und der Wert an dieser Adresse wird in `a` gespeichert.

### Aufgabe 2: Speicherzugriff

Die Variablen `a`, `b`, `c` sind in den Registern `%eax`, `%ebx`, `%ecx` abgelegt. Zusätzlich befindet sich ein Array an der Speicheradresse des Labels `feld` mit 5 Datenworten a 32-Bit, dies entspricht also folgender Java-Deklaration:

```
int[] feld = new int[5];
```

- (a) Geben Sie zwei Möglichkeiten an, wie ein solches Array in Assembler angelegt und mit den Werten (1,2,3,4,5) initialisiert werden kann.
- (b) Wandeln Sie die folgenden Befehle in IA32-Assembler um. Sie brauchen keine Bereichsprüfung des Arrays durchzuführen, beachten Sie jedoch wieder, dass die Werte und Variablen als long anzusehen sind.
- `a = feld[3];`
  - `feld[2] = b;`  
Geben Sie zwei Zugriffsmöglichkeiten auf `feld[2]` an.
  - `feld[c] = feld[b] + feld[feld[0]];`

### Aufgabe 3: Rekursion

Die Funktion  $sum : \mathbb{N} \rightarrow \mathbb{N}$  sei wie folgt definiert.

$$sum(n) = \begin{cases} 0 & \text{falls } n = 0 \\ sum(n-1) + n & \text{sonst} \end{cases}$$

Schreiben Sie ein rekursives Assemblerprogramm, das die Funktion  $sum$  implementiert. Die Parameterübergabe beim Aufruf erfolgt über den Stack. Die Ergebnisrückgabe erfolgt

- (a) über ein Register.
- (b) über den Stack.

### Aufgabe 4: Endianness

- Stellen Sie die Zahl  $1\,123\,581\,321_{(10)}$  als 32-Bit-Zahl in Big-Endian- und Little-Endian-Darstellung dar. Machen Sie jeweils die Reihenfolge der einzelnen Bytes im Speicher kenntlich. Geben Sie anschließend die beiden Darstellungen als Hexadezimalzahlen an.
- Ist  $\frac{00001100}{31} \frac{10111111}{24} \frac{10111111}{23} \frac{00010011}{16} \frac{10111111}{15} \frac{10111111}{8} \frac{10111111}{7} \frac{10111111}{0}$  die Little- oder Big-Endian-Darstellung von  $213849023_{(10)}$ ?

## Hausaufgabe 1: Sudanfunktion

(7 Punkte)

Die rekursiv berechenbare Sudanfunktion wurde 1927 von dem rumänischen Mathematiker Gabriel Sudan publiziert. Sie ist wie folgt definiert<sup>1</sup>.

Für  $x, y, n \in \mathbb{N} \cup \{0\}$  gilt:

$$\begin{aligned}F_0(x, y) &= x + y, \\F_{n+1}(x, 0) &= x, n \geq 0, \\F_{n+1}(x, y + 1) &= F_n(F_{n+1}(x, y), F_{n+1}(x, y) + y + 1), n \geq 0\end{aligned}$$

Machen Sie sich klar, dass die folgende C-Funktion die Sudanfunktion implementiert:

```
int sudan(int n, int x, int y)
{
    if (n == 0) return x + y;
    if (y == 0) return x;

    int t1 = sudan(n, x, y - 1);
    int t2 = sudan(n - 1, t1, t1 + y);

    return t2;
}
```

- (a) Schreiben Sie bitte ein *rekursives* IA32-Assemblerprogramm, das die Sudanfunktion in Abhängigkeit von  $n$ ,  $x$  und  $y$  berechnet. Diese Variablen sollen im `.data`-Bereich definiert sein. Die Parameterübergabe beim Aufruf erfolgt über den Stack. Etwaige Registerüberläufe oder negative Parameter können vernachlässigt werden. Kommentieren Sie Ihren Code ausführlich!

Bitte benutzen Sie folgendes Codegerüst, das Ihnen schon aus der zweiten Übung bekannt ist:

```
.data
intout: .string "Wert %d\n"
n: .long 1
x: .long 2
y: .long 4
.text

.globl main

main:

# Wert im %eax ausgeben
pushl %eax
pushl $intout
call printf
addl $8, %esp

# Exit
movl $1, %eax
```

---

<sup>1</sup>vgl. etwa <https://de.wikipedia.org/wiki/Sudanfunktion>

---

---

```
int $0x80
```

## Hausaufgabe 2: Arithmetische Ausdrücke und der Stack (3 Punkte)

Gegeben ist folgender arithmetischer Ausdruck  $\frac{(b-a)*(e+f)}{a+(b+c)*d}$ .

- (a) Stellen Sie den Ausdruck in LR-Postorder-Darstellung dar. Es ist hilfreich den Ausdruck dazu vorher in einem Baum darzustellen.
- (b) Fügen Sie nun analog zum Vorgehen in der Vorlesung in die LR-Postorder-Darstellung Pfeile nach unten ( $\downarrow$ ) für Push-Befehle und Pfeile nach oben ( $\uparrow$ ) für Pop-Befehle ein, sodass der Ausdruck in Assembler nur mit Hilfe von zwei Registern und dem Stack berechnet werden kann.
- (c) Implementieren Sie nun bitte die Rechnung in IA-32 Assembler. Benutzen Sie ausschließlich die Register `%eax` und `%ebx`. Die Variablen  $a, b, c, d, e, f$  sollen im `.data` Bereich deklariert werden.

Wie immer gilt: **Ihr Code muss auf den Rechnern im RBG-Pool lauffähig sein.**

Sie müssen den Code sowohl bei Ihrem Tutor in der Übung abgeben (gedruckter Code ist erlaubt), als auch im Moodle unter der entsprechenden Abgabe hochladen.